Questions and answers - Saul Alonso Lecture

The following questions were submitted through Google Form. Some may have been answered in the Q&A session already. Nevertheless, we request our lecturers to provide written answers here for the benefit of those who could not attend that session. Thank you!

Slide 22. Where do the elements in X come from ?

X is the matrix that represents the result of applying an embedding layer to the input sequence. The embedding layer converts each item from the input sequence into a higher dimensional vector. In this case, the input sequence is "I am a student," and the resulting embedding matrix (shape: #items x embedding_size) is shown on the left-bottom side of the slide.

Slide 24. What could be the size since simulations can generate tons of it

The appropriate size of the training dataset depends on factors such as model complexity, task difficulty, and the computing resources available. Generally, a larger training dataset might be needed for more complex tasks, but there is no fixed rule; finding the right dataset size involves a trade-off between capturing patterns and practical manageability. As you said, in HEP we can generate almost “infinitely large” simulations, so I advise starting with a relatively small training sample and eventually increasing its size if the model overfits.

Slide 27. Generative models are faster than MC to simulate particle interactions. Can you give some examples of performance differences?

Generative models are typically much faster than current simulations. For example, the following study reports a speedup of a generative model of more than six orders of magnitude compared to the standard simulation:

https://iopscience.iop.org/article/10.1088/1742-6596/1085/2/022005/pdf

Slide 32. Adversarial training when used in an unsupervised way, is it based on the physics informed NN?

There is a technique named domain-adversarial neural network (DANN) used to address domain adaptation issues. DANNs leverage adversarial training to learn

domain-invariant representations from the source domain (simulation, in experimental physics) in an unsupervised way while simultaneously trying to confuse a domain discriminator to distinguish between the source and target (detector data) domains.

Examples in neutrino physics (see also Leigh's lecture):

https://doi.org/10.1088/1748-0221/13/11/P11020
http://dx.doi.org/10.1103/PhysRevD.105.112009

Slide not specified. There are some papers on implicit bias/regularization effect from the gradient-based optimization. Have you seen any papers on implicit regularization effect from the structure of CNN?

As you said, regularization on the optimization algorithm is well-studied (e.g., weight decay). Explicit regularization on the CNN architecture is also standard (e.g., dropout, batch normalization, global average pooling). On the other hand, implicit regularization coming from the CNN structure is a bit less trivial. Still, there was a paper at ICML last year that studied it and found a clear tendency toward locality:

https://arxiv.org/abs/2201.11729

Slide 33. Why are generative models more efficient in terms of simulation than the traditional way?

The generative models shown are neural networks. Thus, to generate samples equivalent to traditional simulations, only the forward pass must be run, which consists of a number of matrix multiplications that can be efficiently run (even in batches) on modern hardware. In contrast, Monte Carlo simulations usually require more memory and are challenging to parallelize, making them time-consuming and computationally expensive. As I pointed out in a previous response, generative models can exhibit a speedup of several orders of magnitude compared to standard simulations (https://iopscience.iop.org/article/10.1088/1742-6596/1085/2/022005/pdf).

Slide 14. How can the numbers of layers in different ResNet models be determined?

The original ResNet implementations have a number in the name indicating the number of convolutional layers of the network. If I understand the question correctly, you ask how to determine the number of layers needed to approach a particular problem. More layers mean more parameters and, thus, more model capacity. In theory, deeper networks are better suited to solve complex problems but are also slower and require

more computing resources. As I mentioned earlier, there is no fixed rule, and it would require experimentation to find the best compromise.

Slide 32. Efforts to quantify uncertainty for classification and regression tasks in ML algorithms have been growing in the ML community but in the field of computer vision, especially in generative models, are there similar efforts in quantifying uncertainty? I imagine it's difficult to quantify something like stable diffusion models generating "Pikachu eating at the Eiffel Tower" images: with such variety of possible solutions, is it even possible to quantify such a distribution of images?

Quantifying the uncertainty of generative models indeed represents a challenging task in computer vision. Generative models lack an objective loss function that estimates the quality of the generated data. This is a nice study that analyzes different evaluation methods in GANs:

https://arxiv.org/abs/1802.03446

That being said, quantifying such uncertainties in HEP is much easier since we train on simulated data. Thus, we know much more accurately what the generated data should look like, so one can easily compare distributions of true (simulated) and generated data for the same input physical parameters (assuming a conditional GAN or a diffusion model is used). Examples:

https://arxiv.org/pdf/2210.09767.pdf

Slide 32. For the uncertainty estimation, do you think out-of-distribution detection in CV like ODIN (Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks) will work in physics project?

Yes, absolutely. I am convinced methods like ODIN (paper: https://arxiv.org/abs/1706.02690) should be explored more in different domains, including physics, where detecting examples out of distribution may help us detect uncertainties and find new physics by identifying data points that deviate from expected patterns.